Should I test/review my?

A. Ten *most-complex* functions

B. Ten *largest* functions

C. Ten *most-fixed* functions

D. Ten *most-discussed* functions...

©A. Hassan
The Secret for Software Decision Making
Hire Experts!
Look through your software data
An international effort to make software repositories actionable

http://msrconf.org

http://promisedata.org

© A. Hassan
Mining Software Repositories (MSR)

- Transforms static record-keeping repositories to **active** repositories
- Makes repository data **actionable** by uncovering hidden **patterns** and **trends**
Natural Language (NL) Artifacts are Pervasive

Source Control
CVS/SVN

Bugzilla
Mailing lists

Field Repos

Crash Repos
Runtime Repos

Code Repos
Sourceforge
GoogleCode

Historical Repositories

Artifacts are Pervasive

©A. Hassan
MSR researchers analyze and cross-link repositories

fixed bug

Bugzilla  Mailinglists  CVS/SVN  Crashes

discussions  Buggy change & Fixing change  Field crashes

New Bug Report
Estimate fix effort
Mark duplicates
Suggest experts and fix!
MSR researchers analyze and cross-link repositories

MSR researchers analyze and cross-link repositories

New Change

Suggest APIs

Warn about risky code or bugs

©A. Hassan
NL Software Artifacts are of Many *Types*

- requirements documents
- code comments
- identifier names
- commit logs
- release notes
- bug reports
- …

- emails discussing bugs, designs, etc.
- mailing list discussions
- test plans
- project websites & wikis
- …
NL Software Artifacts are of Large *Quantity*

- code comments:
  - 2M in Eclipse, 1M in Mozilla, 1M in Linux
- identifier names:
  - 1M in Chrome
- commit logs:
  - 222K for Linux (05-10), 31K for PostgreSQL
- bug reports:
  - 641K in Mozilla, 18K in Linux, 7K in Apache
- ...

**NL data contains useful information, much of which is not in structured data.**
NL Data Contains Useful Info – Example 1

Code comments contain *Specifications*

```c
linux/drivers/scsi/in2000.c:
static int in2000_bus_reset(...) {
    ...
}
```

```c
linux/drivers/scsi/in2000.c:
static int reset_hardware(...) {
    ...
}
```

Tan et al. “/*iComment: Bugs or Bad Comments?*/”, SOSP’07

No lock acquisition ⇒ A bug!

```c
reset_hardware(...);
...
```
Using Comments for Better Software Quality

• Specifications/rules (examples from real-world software):
  – Calling context: /* Must be called with interrupts disabled */
  – Calling order: /* Call scsi_free before mem_free since ... */
  – Unit: int mem; /* memory in 128 MB units */
  – Help ensure correct software evolution:
    /* WARNING: If you change any of these defines, make sure to change ... */
    /* FIXME: We should group addresses here. */

• Beyond reliability:
  – Help reduce code navigation:
    /* See comment in struct sock definition to understand ... */
    timeout_id_t msd_timeout_id; /* id returned by timeout() */
API documentation contains resource usages

- `java.sql.ResultSet.deleteRow()` : “Deletes the current row from this `ResultSet` object and from the underlying database”
- `java.sql.ResultSet.close()` : “Releases this `ResultSet` object’s database and JDBC resources immediately instead of waiting for this to happen when it is automatically closed”.

```
java.sql.ResultSet.deleteRow() ➔
java.sql.ResultSet.close()
```
Don’t ignore the semantics of identifiers

```java
noFirewall = new JRadioButton("No firewall or proxy");
socksFirewall = new JRadioButton("SOCKS 4/5 Firewall");
webProxy = new JRadioButton("HTTP Web Proxy");

allButtons = new ButtonGroup();
allButtons.add(socksFirewall);
allButtons.add(webProxy);
allButtons.add(noFirewall);
socksFirewall.addActionListener(rad);
webProxy.addActionListener(rad);
noFirewall.addActionListener(rad);
```

Create RadioButtons

Add RadioButtons to allButtons

Add radioActionListener to RadioButtons
Challenges in Analyzing NL Data

• Unstructured
  – Hard to parse, sometimes wrong grammar

• Ambiguous: often has no defined or precise semantics (as opposed to source code)
  – Hard to understand

• Many ways to represent similar concepts
  – Hard to extract information from

/* We need to acquire the write IRQ lock before calling ep_unlink(). */
/* Lock must be acquired on entry to this function. */
/* Caller must hold instance lock! */
Why Analyzing NL Data is Easy(?)

• Redundant data
• Easy to get “good” results for simple tasks
  – Simple algorithms without much tuning effort
• Evolution/version history readily available
• Many techniques to borrow from text analytics: NLP, Machine Learning (ML), Information Retrieval (IR), etc.
Text Analytics

Knowledge Rep. & Reasoning / Tagging

Semantic Web

Web 2.0

Computational Linguistics

Natural Language Processing

Information Retrieval

Data Analysis

Search & DB

Machine Learning

Text Mining

©M. Grobelnik, D. Mladenic
Why Analyzing NL Data is Hard(?)

• Domain specific words/phrases, and meanings
  – “Call a function” vs. call a friend
  – “Computer memory” vs. human memory
  – “This method also returns false if path is null”

• Poor quality of text
  – Inconsistent
  – grammar mistakes
    • “true if path is an absolute path; otherwise false” for the File class in .NET framework
  – Incomplete information
Some Major NLP/Text Analytics Tools

Text Miner

Stanford Parser
http://nlp.stanford.edu/software/lex-parser.shtml

http://nlp.stanford.edu/links/statnlp.html
http://www.kdnuggets.com/software/text.html

Text Analytics for Surveys

http://uima.apache.org/
Outline

• Motivation
  – Why mining NL data in software engineering?
  – Opportunities and Challenges

• Popular text analytics techniques
  – Sample research work

• Future directions
Dimensions in Text Analytics

• Three major dimensions of text analytics:
  – Representations
    • ...from words to partial/full parsing
  – Techniques
    • ...from manual work to learning
  – Tasks
    • ...from search, over (un-)supervised learning, summarization, ...
Major Text Representations

• Words (stop words, stemming)
• Part-of-speech tags

• Chunk parsing (chunking)
• Semantic role labeling
• Vector space model

©M. Grobelnik, D. Mladenic
Words’ Properties

• Relations among word surface forms and their senses:
  – **Homonymy**: same form, but different meaning (e.g. bank: river bank, financial institution)
  – **Polysemy**: same form, related meaning (e.g. bank: blood bank, financial institution)
  – **Synonymy**: different form, same meaning (e.g. singer, vocalist)
  – **Hyponymy**: one word denotes a subclass of another (e.g. breakfast, meal)

• General thesaurus: WordNet, existing in many other languages (e.g. EuroWordNet)
Stop Words

• Stop words are words that from non-linguistic view do not carry information
  – ...they have mainly functional role
  – ...usually we remove them to help mining techniques to perform better

• Stop words are language dependent – examples:
  – **English**: A, ABOUT, ABOVE, ACROSS, AFTER, AGAIN, AGAINST, ALL, ALMOST, ALONE, ALONG, ALREADY, ...
Stemming

• Different forms of the same word are usually problematic for text analysis, because they have different spelling and similar meaning (e.g. learns, learned, learning,...)

• Stemming is a process of transforming a word into its stem (normalized form)
  —...stemming provides an inexpensive mechanism to merge
Stemming cont.

• For English is mostly used Porter stemmer at http://www.tartarus.org/~martin/PorterStemmer/

• Example cascade rules used in English Porter stemmer
  – ATIONAL -> ATE           relational -> relate
  – TIONAL -> TION           conditional -> condition
  – ENCI       -> ENCE       valenci -> valence
  – ANCI       -> ANCE       hesitanci -> hesitance
  – IZER       -> IZE        digitizer -> digitize
  – ABLI       -> ABLE       conformabli -> conformable
  – ALLI       -> AL         radicalli -> radical
  – ENTLI      -> ENT        differentli -> different
  – ELI        -> E          vileli -> vile
  – OUSLI      -> OUS        analogousli -> analogous
Part-of-Speech Tags

• Part-of-speech tags specify word types enabling to differentiate words functions
  – For text analysis, part-of-speech tag is used mainly for “information extraction” where we are interested in e.g., named entities (“noun phrases”)
  – Another possible use is reduction of the vocabulary (features)
    • ...it is known that nouns carry most of the information in text documents

• Part-of-Speech taggers are usually learned on manually tagged data
# Part-of-Speech Table

<table>
<thead>
<tr>
<th>part of speech</th>
<th>function or &quot;job&quot;</th>
<th>example words</th>
<th>example sentences</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Verb</strong></td>
<td>action or state</td>
<td>(to) be, have, do, like, work, sing, can, must</td>
<td>EnglishClub.com is a web site. I like EnglishClub.com.</td>
</tr>
<tr>
<td><strong>Noun</strong></td>
<td>thing or person</td>
<td>pen, dog, work, music, town, London, teacher, John</td>
<td>This is my dog. He lives in my house. We live in London.</td>
</tr>
<tr>
<td><strong>Adjective</strong></td>
<td>describes a noun</td>
<td>a/an, the, 69, some, good, big, red, well, interesting</td>
<td>My dog is big. I like big dogs.</td>
</tr>
<tr>
<td><strong>Adverb</strong></td>
<td>describes a verb, adjective or adverb</td>
<td>quickly, silently, well, badly, very, really</td>
<td>My dog eats quickly. When he is very hungry, he eats really quickly.</td>
</tr>
<tr>
<td><strong>Pronoun</strong></td>
<td>replaces a noun</td>
<td>I, you, he, she, some</td>
<td>Tara is Indian. She is beautiful.</td>
</tr>
<tr>
<td><strong>Preposition</strong></td>
<td>links a noun to another word</td>
<td>to, at, after, on, but</td>
<td>We went to school on Monday.</td>
</tr>
<tr>
<td><strong>Conjunction</strong></td>
<td>joins clauses or sentences or words</td>
<td>and, but, when</td>
<td>I like dogs and I like cats. I like dogs but I don't like cats.</td>
</tr>
<tr>
<td><strong>Interjection</strong></td>
<td>short exclamation, sometimes inserted into a sentence</td>
<td>oh!, ouch!, hi!, well</td>
<td>Ouch! That hurts! Hi! How are you? Well, I don't know.</td>
</tr>
</tbody>
</table>

Part-of-Speech Examples

<table>
<thead>
<tr>
<th>verb</th>
<th>noun</th>
<th>verb</th>
<th>noun</th>
<th>verb</th>
<th>verb</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stop!</td>
<td>John</td>
<td>works.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>pronoun</th>
<th>verb</th>
<th>noun</th>
</tr>
</thead>
<tbody>
<tr>
<td>She</td>
<td>loves</td>
<td>animals.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>noun</th>
<th>verb</th>
<th>adjective</th>
<th>noun</th>
</tr>
</thead>
<tbody>
<tr>
<td>Animals</td>
<td>like</td>
<td>kind</td>
<td>people.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>noun</th>
<th>verb</th>
<th>noun</th>
<th>adverb</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tara</td>
<td>speaks</td>
<td>English</td>
<td>well.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>pronoun</th>
<th>verb</th>
<th>preposition</th>
<th>adjective</th>
<th>noun</th>
<th>adverb</th>
</tr>
</thead>
<tbody>
<tr>
<td>She</td>
<td>ran</td>
<td>to</td>
<td>the</td>
<td>station</td>
<td>quickly.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>pron.</th>
<th>verb</th>
<th>adj.</th>
<th>noun</th>
<th>conjunction</th>
<th>pron.</th>
<th>verb</th>
<th>pron.</th>
</tr>
</thead>
<tbody>
<tr>
<td>She</td>
<td>likes</td>
<td>big</td>
<td>snakes</td>
<td>but</td>
<td>I</td>
<td>hate</td>
<td>them.</td>
</tr>
</tbody>
</table>

Here is a sentence that contains every part of speech:

<table>
<thead>
<tr>
<th>interjection</th>
<th>pron.</th>
<th>conj.</th>
<th>adj.</th>
<th>noun</th>
<th>verb</th>
<th>prep.</th>
<th>noun</th>
<th>adverb</th>
</tr>
</thead>
<tbody>
<tr>
<td>Well,</td>
<td>she</td>
<td>and</td>
<td>young</td>
<td>John</td>
<td>walk</td>
<td>to</td>
<td>school</td>
<td>slowly.</td>
</tr>
</tbody>
</table>
## Part of Speech Tags

<table>
<thead>
<tr>
<th>Tag</th>
<th>Description</th>
<th>Example</th>
<th>Tag</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>CC</td>
<td>Coordin. Conjunction</td>
<td><em>and, but, or</em></td>
<td>SYM</td>
<td>Symbol</td>
<td><em>+, %, &amp;</em></td>
</tr>
<tr>
<td>CD</td>
<td>Cardinal number</td>
<td><em>one, two, three</em></td>
<td>TO</td>
<td>“to”</td>
<td><em>to</em></td>
</tr>
<tr>
<td>DT</td>
<td>Determiner</td>
<td><em>a, the</em></td>
<td>UH</td>
<td>Interjection</td>
<td><em>ah, oops</em></td>
</tr>
<tr>
<td>EX</td>
<td>Existential ‘there’</td>
<td><em>there</em></td>
<td>VB</td>
<td>Verb, base form</td>
<td><em>eat</em></td>
</tr>
<tr>
<td>FW</td>
<td>Foreign word</td>
<td><em>mea culpa</em></td>
<td>VBD</td>
<td>Verb, past tense</td>
<td><em>ate</em></td>
</tr>
<tr>
<td>IN</td>
<td>Preposition/sub-conj</td>
<td><em>of, in, by</em></td>
<td>VBG</td>
<td>Verb, gerund</td>
<td><em>eating</em></td>
</tr>
<tr>
<td>JJ</td>
<td>Adjective</td>
<td><em>yellow</em></td>
<td>VBN</td>
<td>Verb, past participle</td>
<td><em>eaten</em></td>
</tr>
<tr>
<td>JJR</td>
<td>Adj., comparative</td>
<td><em>bigger</em></td>
<td>VBP</td>
<td>Verb, non-3sg pres</td>
<td><em>eat</em></td>
</tr>
<tr>
<td>JJR</td>
<td>Adj., superlative</td>
<td><em>wildest</em></td>
<td>VBZ</td>
<td>Verb, 3sg pres</td>
<td><em>eats</em></td>
</tr>
<tr>
<td>LS</td>
<td>List item marker</td>
<td><em>1, 2, One</em></td>
<td>WDT</td>
<td>Wh-determiner</td>
<td><em>which, that</em></td>
</tr>
<tr>
<td>MD</td>
<td>Modal</td>
<td><em>can, should</em></td>
<td>WP</td>
<td>Wh-pronoun</td>
<td><em>what, who</em></td>
</tr>
<tr>
<td>NN</td>
<td>Noun, sing. or mass</td>
<td><em>llama</em></td>
<td>WPS</td>
<td>Possessive wh-</td>
<td><em>whose</em></td>
</tr>
<tr>
<td>NNS</td>
<td>Noun, plural</td>
<td><em>llamas</em></td>
<td>WRB</td>
<td>Wh-adverb</td>
<td><em>how, where</em></td>
</tr>
<tr>
<td>NNP</td>
<td>Proper noun, singular</td>
<td><em>IBM</em></td>
<td>$</td>
<td>Dollar sign</td>
<td><em>$</em></td>
</tr>
<tr>
<td>NNPS</td>
<td>Proper noun, plural</td>
<td><em>Carolinatas</em></td>
<td>#</td>
<td>Pound sign</td>
<td><em>#</em></td>
</tr>
<tr>
<td>PDT</td>
<td>Predeterminer</td>
<td><em>all, both</em></td>
<td>”</td>
<td>Left quote</td>
<td><em>‘ or “</em></td>
</tr>
<tr>
<td>POS</td>
<td>Possessive ending</td>
<td><em>’s</em></td>
<td>”</td>
<td>Right quote</td>
<td><em>‘ or ”</em></td>
</tr>
<tr>
<td>PP</td>
<td>Personal pronoun</td>
<td><em>I, you, he</em></td>
<td>(</td>
<td>Left parenthesis</td>
<td><em>[, (, {, &lt;)</em></td>
</tr>
<tr>
<td>PPS</td>
<td>Possessive pronoun</td>
<td><em>your, one’s</em></td>
<td>)</td>
<td>Right parenthesis</td>
<td>*], ),</td>
</tr>
<tr>
<td>RB</td>
<td>Adverb</td>
<td><em>quickly, never</em></td>
<td>.</td>
<td>Comma</td>
<td><em>,</em></td>
</tr>
<tr>
<td>RBR</td>
<td>Adverb, comparative</td>
<td><em>faster</em></td>
<td>;</td>
<td>Sentence-final punc</td>
<td><em>(. ! ?)</em></td>
</tr>
<tr>
<td>RBS</td>
<td>Adverb, superlative</td>
<td><em>fastest</em></td>
<td>:</td>
<td>Mid-sentence punc</td>
<td><em>(; ; ... --)</em></td>
</tr>
<tr>
<td>RP</td>
<td>Particle</td>
<td><em>up, off</em></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

[http://www2.sis.pitt.edu/~is2420/class-notes/2.pdf](http://www2.sis.pitt.edu/~is2420/class-notes/2.pdf)
Full Parsing

- Parsing provides maximum structural information per sentence
- Input: a sentence → output: a parse tree
- For most text analysis techniques, the information in parse trees is too complex

Problems with full parsing:
- Low accuracy
- Slow
- Domain Specific
Chunk Parsing

• Break text up into non-overlapping contiguous subsets of tokens.
  – aka. partial/shallow parsing, light parsing.

• What is it useful for?
  – Entity recognition
    • people, locations, organizations
  – Studying linguistic patterns
    • gave NP
    • gave up NP in NP
    • gave NP NP
    • gave NP to NP
  – Can ignore complex structure when not relevant
Chunk Parsing

Goal: divide a sentence into a sequence of chunks.

• Chunks are non-overlapping regions of a text
  
  [I] saw [a tall man] in [the park]

• Chunks are non-recursive
  – A chunk cannot contain other chunks

• Chunks are non-exhaustive
  – Not all words are included in the chunks
Chunk Parsing Techniques

• Chunk parsers usually ignore lexical content
• Only need to look at part-of-speech tags

• Techniques for implementing chunk parsing
  – E.g., Regular expression matching
Regular Expression Matching

• Define a regular expression that matches the sequences of tags in a chunk
  – A simple noun phrase chunk regrexp:
    \(<DT>\ ? <JJ>\ * <NN.?>\)

• Chunk all matching subsequences:
The /DT little /JJ cat /NN sat /VBD on /IN the /DT mat /NN
[The /DT little /JJ cat /NN] sat /VBD on /IN [the /DT mat /NN]

• If matching subsequences overlap, the first one gets priority

DT: Determinner   JJ: Adjective   NN: Noun, sing, or mass
VBD: Verb, past tense   IN: Prepostion/sub-conj  Verb
Semantic Role Labeling
Giving Semantic Labels to Phrases

• \([\text{AGENT John}] \text{ broke} [\text{THEME the window}]\)

• \([\text{THEME The window}] \text{ broke}\)

• \([\text{AGENT Sotheby’s}] \text{ .. offered} [\text{RECIPIENT the Dorrance heirs}] [\text{THEME a money-back guarantee}]\)

• \([\text{AGENT Sotheby’s}] \text{ offered} [\text{THEME a money-back guarantee}] \text{ to} [\text{RECIPIENT the Dorrance heirs}]\)

• \([\text{THEME a money-back guarantee}] \text{ offered by} [\text{AGENT Sotheby’s}]\)

• \([\text{RECIPIENT the Dorrance heirs}] \text{ will} [\text{ARM-NEG not}] \text{ be offered} [\text{THEME a money-back guarantee}]\)
Semantic Role Labeling Good for Question Answering

Q: What was the name of the first computer system that defeated Kasparov?
A: [PATIENT Kasparov] was defeated by [AGENT Deep Blue] [TIME in 1997].

Q: When was Napoleon defeated?

Look for: [PATIENT Napoleon] [PRED defeat-synset] [ARGM-TMP *ANS*]

More generally:

- Who hit Scott with a baseball?
- Whom did Kristina hit with a baseball?
- What did Kristina hit Scott with?
- When did Kristina hit Scott with a baseball?
## Typical Semantic Roles

<table>
<thead>
<tr>
<th>Thematic Role</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>AGENT</td>
<td>The volitional causer of an event</td>
</tr>
<tr>
<td>EXPERINER</td>
<td>The experiencer of an event</td>
</tr>
<tr>
<td>FORCE</td>
<td>The non-volitional causer of the event</td>
</tr>
<tr>
<td>THEME</td>
<td>The participant most directly affected by an event</td>
</tr>
<tr>
<td>RESULT</td>
<td>The end product of an event</td>
</tr>
<tr>
<td>CONTENT</td>
<td>The proposition or content of a propositional event</td>
</tr>
<tr>
<td>INSTRUMENT</td>
<td>An instrument used in an event</td>
</tr>
<tr>
<td>BENEFICIARY</td>
<td>The beneficiary of an event</td>
</tr>
<tr>
<td>SOURCE</td>
<td>The origin of the object of a transfer event</td>
</tr>
<tr>
<td>GOAL</td>
<td>The destination of an object of a transfer event</td>
</tr>
</tbody>
</table>
## Example Semantic Roles

<table>
<thead>
<tr>
<th>Thematic Role</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>AGENT</td>
<td><em>The waiter</em> spilled the soup.</td>
</tr>
<tr>
<td>EXPERIENCER</td>
<td><em>John</em> has a headache.</td>
</tr>
<tr>
<td>FORCE</td>
<td><em>The wind</em> blows debris from the mall into our yards.</td>
</tr>
<tr>
<td>THEME</td>
<td>Only after Benjamin Franklin broke <em>the ice</em>...</td>
</tr>
<tr>
<td>RESULT</td>
<td>The French government has built a <em>regulation-size baseball diamond</em>...</td>
</tr>
<tr>
<td>CONTENT</td>
<td>Mona asked “<em>You met Mary Ann at a supermarket</em>?”</td>
</tr>
<tr>
<td>INSTRUMENT</td>
<td>He turned to poaching catfish, stunning them with a <em>shocking device</em>...</td>
</tr>
<tr>
<td>BENEFICIARY</td>
<td>Whenever Ann Callahan makes hotel reservations <em>for her boss</em>...</td>
</tr>
<tr>
<td>SOURCE</td>
<td>I flew in <em>from Boston</em>.</td>
</tr>
<tr>
<td>GOAL</td>
<td>I drove to <em>Portland</em>.</td>
</tr>
</tbody>
</table>
Inferring Resource Specifications

• Named entity recognition with chunk tagger
• Training

Tagged method descriptions

• Tagging

Zhong et al. Inferring Resource Specifications from Natural Language API Documentation, ASE’09
Resource Specification Example

• Action-resource pairs
  – `createInteraction()`: `<create, connection> ← “Creates an interaction associated with this connection.”`
  – `getMetaData()`: `<get, connection> ← “Gets the information on the underlying EIS instance represented through an active connection.”`
  – `close()`: `<close, connection> ← “Initiates close of the connection handle at the application level.”`

• Inferred resource specification

Zhong et al. Inferring Resource Specifications from Natural Language API Documentation, ASE’09
### iComment: Rule Template Examples

<table>
<thead>
<tr>
<th>ID</th>
<th>Rule Template Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><code>&lt;Lock L&gt;</code> must be held before entering <code>&lt;Function F&gt;</code>.</td>
</tr>
<tr>
<td>1</td>
<td><code>&lt;Lock L&gt;</code> must NOT be held before entering <code>&lt;Function F&gt;</code>.</td>
</tr>
<tr>
<td>2</td>
<td><code>&lt;Lock L&gt;</code> must be held in <code>&lt;Function F&gt;</code>.</td>
</tr>
<tr>
<td>2</td>
<td><code>&lt;Lock L&gt;</code> must NOT be held in <code>&lt;Function F&gt;</code>.</td>
</tr>
<tr>
<td>3</td>
<td><code>&lt;Function A&gt;</code> must be called from <code>&lt;Function B&gt;</code></td>
</tr>
<tr>
<td>3</td>
<td><code>&lt;Function A&gt;</code> must NOT be called from <code>&lt;Function B&gt;</code></td>
</tr>
</tbody>
</table>

- L, F, A and B are rule parameters.
- Many other templates can be added.

/* We need to **acquire** the write IRQ lock **before** calling `ep_unlink()` */

/* Lock must be **acquired on entry** to this function. */

/* **Caller** must **hold** instance lock! */
iComment: Extracting Target Comments

- **Topic keyword filtering**
- **Correlated word filtering**

Clustering and simple statistics to mine topic keywords and correlated words

<table>
<thead>
<tr>
<th></th>
<th>Linux</th>
<th>hold</th>
<th>acquire</th>
<th>call</th>
<th>unlock</th>
<th>protect</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mozilla</td>
<td>hold</td>
<td>acquire</td>
<td>unlock</td>
<td>protect</td>
<td>call</td>
</tr>
</tbody>
</table>

Take lock as the topic:

*A: /* return -EBUSY if a lock is held. */
*B: /* Lock must be held on entry to this function. */
*C: /* Caller must acquire instance lock! */

*D: /* Mutex locked flags */

...
Vector Space Model

<table>
<thead>
<tr>
<th></th>
<th>away</th>
<th>brings</th>
<th>here</th>
<th>is</th>
<th>key</th>
<th>man</th>
<th>not</th>
<th>that</th>
<th>the</th>
<th>with</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>y</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Major Text Analytics Tasks

• Duplicate-Document Detection
• Document Summarization
• Document Categorization
• Document Clustering
• Document Parsing
Duplicate-Document Detection

- **Task**: the task is to select duplicate documents among a set of documents D for the given document d

- **Basic approach**:
  - Compare the similarity between d and each \( d_i \) in D
  - Rank all \( d_i \) in D with similarity \( \geq \) threshold based on similarity
Bugzilla Bug Report

Bugzilla Bug 337641

**Bug List:** (54 of 71) **First Last Prev Next**  **Show last search results**  **Search page**  **Enter new bug**

**Bug#:** 337641  **alias:**

**Product:** Firefox  
**Component:** Download Manager  
**Status:** NEW  
**Resolution:** bill@firefox.org  
**Assigned To:** Nobody is working on this, feel free to take it  
<nobody@mozilla.org>

**Hardware:** PC  
**OS:** Windows 2000  
**Version:** Trunk  
**Priority:** —  
**Severity:** normal  
**Target Milestone:**

**QA Contact:** download.manager@firefox.bugs

**URL:**

**Summary:** download doesn't start right away anymore

**Status Whiteboard:**

**Keywords:** regression

©J. Anvik
Sample Bugzilla Bug Report

Bugzilla Bug 338009

Browser Crashes at cbs.com

Last modified: 2006-05-15 09:27:44 PDT

Bug List: (15 of 37) First Last Prev Next Show last search results Search page Enter new bug

Bug#: 338009 alias:
Product: Firefox
Component: General
Status: UNCONFIRMED

Description: [reply]
Each time I visit http://www.cbs.com/, Firefox crashes before the page is loaded. I can tell what element of the page is crashing the browser though.

Reproducible: Always
Steps to Reproduce:
1. Open Browser
2. Enter http://www.cbs.com/
3. Press return

Actual Results:
Page starts to load, and then crashes.

Expected Results:
The browser doesn't crash.

No other sites so far have displayed this behavior.

Duplicate?
Reproducible?
Assigned To: ?

Bugzilla: open source bug tracking tool
http://www.bugzilla.org/

A. E. Hassan and T. Xie: Mining Software Engineering Data
Detecting Duplicate Bug Reports: Workflow

Bug repository

- Retrieve
- Recommend

Triager

- Compare

Suggested List

New report

Approaches based on only NL descriptions
[Runeson et al. ICSE’07, Jalbert & Weimer DSN’08]
An Example in Firefox using only NL information may fail

• Bug-260331: After closing Firefox, the process is still running. Cannot reopen Firefox after that, unless the previous process is killed manually

• Bug-239223: (Ghostproc) – [Meta] firefox.exe doesn't always exit after closing all windows; session-specific data retained
An Example in Firefox using only execution information may fail

- Bug-244372: "Document contains no data" message on continuation page of NY Times article
Mining Both NL and Execution Info

- Calculate natural-language-based similarity
- Calculate execution-information-based similarity
- Combine the two similarities
- Retrieve the most similar bug reports

An Example of Mining Integrated Data of Different Types

Wang et al. An Approach to Detecting Duplicate Bug Reports using Natural Language and Execution Information. ICSE’08
Document Summarization

• **Task**: the task is to produce shorter, summary version of an original document

• Two main approaches to the problem:
  – **Selection based** – summary is selection of sentences from an original document
  – **Knowledge rich** – performing semantic analysis, representing the meaning and generating the text satisfying length restriction

• Summarizing bug reports with conversational structure

Example of selection based approach from MS Word

**Tutorial title**

Text Mining and Link Analysis for Web Data

Presenter contact information including the e-mail address

Dunja Mladenic  
Address: J. Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia  
E-mail: Dunja.Mladenic@ijs.si  
Phone: +386 1 4773 377

Marko Grobelnik  
Address: J. Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia  
E-mail: Marko.Grobelnik@ijs.si  
Phone: +386 1 4773 778

Aims/Learning objectives:

The aim of this tutorial is to present topics from the areas of text mining and link analysis in the relationship to the web data. The goal is to show the whole list of nontrivial problems appearing in everyday life and occasionally in professional work with the web and to show how they can be approached using text mining and link analysis techniques and tools. The goal is to make an overview of the available approaches, which are potentially useful for solving interesting problems connected to the documents and their linkage coming from the web structure.

Duration (half or full day)

Half day, but it could be scaled to full day

Scope (general topic area) and why it is relevant for WWW2004:

The tutorial’s relevance for the WWW2004 is in the presentation of analytic approaches used on the web data (text+links). In particular, the tutorial will focus on the possibilities offered by two very active and relevant subfields of data mining: text mining and link analysis. The relevance of these topics to the WWW2004 public is in extending possible activities, which could be used in shaping, understanding, and potentially predicting the static and dynamic nature of the web. Analysis of such data offers typically new insights in the nature of the complex web data.
Example Bug Report

conversational structure

Bug 188311 - The applet panel should not overlap applets

Product: amarok
Component: ContextView
Priority: NOR
Status: RESOLVED
Severity: wishlist
Resolution: FIXED
Target: ---

Votes: 0

Description From mangus 2009-03-28 11:35:10

In amarok2-svn I like the the new contextview, but I found the new bottom bar for managing applets annoying, as it covers parts of other applets sometimes, like lyrics one, so that you miss a part of it. Could be handy to have it appear and disappear on mouseover.

thanks

------ Comment #1 From Dan 2009-03-28 14:53:55 ------

The real solution is to make it not cover applets, not make it appear/disappear on mouse over.

------ Comment #2 From Leo 2009-03-29 14:34:53 ------

i dont understand your point, dan... how do we make it not cover applets?
Example Extracted Summary of Bug Report

SUMMARY: The applet panel should not overlap applets

In amarok2-svn I like the the new contextview, but I found the new bottom bar for managing applets annoying, as it covers parts of other applets sometimes, like lyrics one, so that you miss a part of it.

The real solution is to make it not cover applets, not make it appear/disappear on mouse over.

i dont understand your point, dan... how do we make it not cover applets?

Applets should not be larger than the viewable area, if there’s an applet above it, then the lower applet should get a smaller sizehint, and resize if necessary when it’s the active applet (and therefore the only one on the screen)

The bug that is being shown here is the fact that you cannot yet resize your applets, and as such we also don’t set default sizes sanely.

Of course :) Just thought i should point out that the feature is not yet completed - the polish that’s gone into it lately could seem like an indication of feature completion, and as such it would seem the prudent course to inform you that that is not the case :)

[Rastkar et al. ICSE’ 10]
Document Categorization

• **Given:** set of documents labeled with content categories

• **The goal:** to build a model which would automatically assign right content categories to new unlabeled documents.

• Content categories can be:
  – unstructured (e.g., Reuters) or
  – structured (e.g., Yahoo, DMOz, Medline)
Document Categorization Workflow

labeled documents → Document Classifier → document category (label) → Machine learning

unlabeled document
Representation of Bug Repository: Buckets

- A hashmap-like data structure
  - Key: master reports
  - Value: corresponding duplicate reports
- Each bucket reports the same defect
- When a new report comes
  - Master? Create a new bucket
  - Otherwise, add it to its bucket

<table>
<thead>
<tr>
<th>Buckets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Master-1: dup-1.1, dup-2.1, ......</td>
</tr>
<tr>
<td>Master-2: dup-2.1, dup-2.2, ......</td>
</tr>
<tr>
<td>Master-3: dup-3.1, dup-3.2, ......</td>
</tr>
<tr>
<td>......</td>
</tr>
<tr>
<td>Master-M: dup-M.1, dup-M.2, ......</td>
</tr>
</tbody>
</table>

Sun et al. A discriminative model approach for accurate duplicate bug report retrieval, ICSE’10
Training Set for Model Learning

• Construct two-class training set
  – Each data instance is a pair of reports
  – **Duplicate** Class: within each bucket, (master, dup), (dup₁, dup₂)
  – **Non-duplicate** Class: pairs, each of which consists of two reports from different buckets

• Learn discriminative model
Applying Models to Retrieve Duplicates

- Retrieve top-$N$ buckets with highest similarity

Sun et al. A discriminative model approach for accurate duplicate bug report retrieval, ICSE’10
Classification of Security Bug Reports

Two bug reports describing a buffer overflow

Security bug report

“An attacker can exploit a buffer overflow by sending excessive data into an input field.”

Mislabeled security bug report

“The system crashes when receiving excessive text in the input field”
### Term-by-document frequency matrix quantifies a document

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Attack</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Buffer Overflow</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Vulnerability</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Label:**
- **Security**
- **Non-Security**
- **?**

Document Clustering

• Clustering is a process of finding natural groups in the data in a unsupervised way (no class labels are pre-assigned to documents)

• Key element is similarity measure
  – In document clustering, cosine similarity is most widely used

• Cluster open source projects
  – Use identifiers (e.g., variable names, function names) as features
    • “gtk_window” represents some window
    • The source code near “gtk_window” contains some GUI operation on the window
  – “gtk_window”, “gtk_main”, and “gpointer” ← GTK related software system

Kawaguchi et al. MUDABlue: An Automatic Categorization System for Open Source. APSEC ’04
Document Parsing: NL Clue Extraction from Source Code

- **Key Challenges:**
  - Decode name usage
  - Develop automatic NL clue extraction process
  - Create NL-based program representation

What was Pete *thinking* when he wrote this code?

Molly, the Maintainer

[Pollock et al. MACS 05, LATE 05] ©L. Pollock
Which NL Clues to Use?

• Software Maintenance
  – Typically focused on actions
  – Objects well-modularized

• Focus on actions
  – Correspond to verbs
  – Verbs need Direct Object (DO)

➤ Extract verb-DO pairs

©L. Pollock [Pollock et al. AOSD 06, IET 08]
Extracting Verb-DO Pairs

Two types of extraction

```
class Player{
    /**
     * Play a specified file with specified time interval
     */
    boolean play(final File file, final float fPosition, final long length) {
        fCurrent = file;
        try {
            playerImpl = null;
            //make sure to stop non-fading players
            stop(false);
            //Choose the player
            Class cPlayer = file.getTrack().getType().getPlayerImpl();
            ...
        }
    }
```
Extracting Clues from Signatures

1. Part-of-speech tag method name
2. Chunk method name
3. Identify **Verb** and **Direct-Object** (DO)

```java
generic UserList getUserListFromFile( String path ) throws IOException {
    try {
        File tmpFile = new File( path );
        return parseFile(tmpFile);
    }
    catch( java.io.IOException e ) {
        throw new IOException( "UserList format issue" + path + " file " + e );
    }
}
```
Representing Verb-DO Pairs

Action-Oriented Identifier Graph

verb1, DO1
verb1, DO2
verb3, DO2
verb2, DO3

©L. Pollock
Action-Oriented Identifier Graph: Example

play, file
play, playlist
remove, playlist
add, listener

source code files

©L. Pollock
Outline

• Motivation
  – Why mining NL data in software engineering?
  – Opportunities and Challenges

• Popular text analytics techniques
  – Sample research work

• Future directions
Software Intelligence: The Future of Mining Software Engineering Data

Ahmed E. Hassan
School of Computing
Queen’s University
Kingston, ON, Canada
ahmed@cs.queensu.ca

Tao Xie
Department of Computer Science
North Carolina State University
Raleigh, NC, USA
xie@csc.ncsu.edu

ABSTRACT
Much of software engineering research revolves around daily decision-making and seeks to support the daily decision-making processes. Software Intelligence (SI) provides practitioners with access to specialized fact-supported views of their software systems so that they can answer critical questions about it. Using SI, owners, maintainers, and developers of software systems can perform long-term and short-term informed strategic planning.

1. INTRODUCTION
Sufficient software engineering data has been accumulated to support an SI initiative. However, data on its own does not provide a framework for realizing SI in the near future. This position paper summarizes the state of practice and research of SI, and lays out future research directions for mining software engineering data to enable SI.

Categories and Subject Descriptors
D.2.7 [Software Engineering]: Maintenance
D.2.2 [Software Engineering]: Design Methodology

General Terms
Documentation, Economics, Experimentation, Management, Measurement, Reliability, Verification

Keywords
Software intelligence, mining software engineering data, mining software repositories

Sufficient software engineering data has been accumulated to support an SI initiative. However, data on its own does not provide a framework for realizing SI in the near future. This position paper summarizes the state of practice and research of SI, and lays out future research directions for mining software engineering data to enable SI.

http://people. engr. ncsu. edu/ txie/ publications/ foser10-si. pdf

• Assist decision making (actionable)
• Assist not just developers
This is too good to be true! Should I sell my dice?!

What is the catch?!

Anyone using this today?
From Business Intelligence to Software Intelligence/Analytics

Get Real

rely on data results alone and declare improvements on benchmarks as “successes”

What do developers think about your result? Is it applicable in their context? How much would it help them in their daily work?

Figure 2: Color-coding keys by their defect correlation; (red = strong). The five strongest co

The IROP keyboard

Zeller et al. Failure is a Four-Letter Word – A Parody in Empirical Research, PROMISE’11
Future Directions

• Make results actionable and engage users to take action on them
• Expand to more software engineering tasks
• Integrate mining of NL data and structured data
• Build/share domain specific NLP/ML tools
  – Task domains, application domains
• Build/share benchmarks for NLP/text mining tasks in SE
• ...
Publishing Advice

• Report the statistical significance of your results:
  – Get a statistics book (one for social scientist, not for mathematicians)

• Discuss any limitations of your findings based on the characteristics of the studied repositories:
  – Make sure you manually examine the repositories. Do not fully automate the process!

• Avoid over-emphasizing contributions of new NLP techniques in submissions to SE venues

• Relevant conferences/workshops:
  – main SE conferences, ICSM, ISSTA, MSR, WODA, ...
Mining Software Repositories

Very active research area in SE

- MSR is the most attended ICSE event in last 7+ yrs
  - [http://msrconf.org](http://msrconf.org)

- Special Issue of IEEE TSE on MSR
  - 15% of all submissions of TSE in 2004
  - Fastest review cycle in TSE history: 8 months

- Special Issues
  - Journal of Empirical Software Engineering
  - Journal of Soft. Maintenance and Evolution
  - IEEE Software (July 1st, 2008)
Submit to MSR 2012

Important Dates

Abstract: Feb 06, 2012
Research/short papers: Feb 10, 2012
Challenge papers: Mar 02, 2012

http://msrconf.org/
More to Come

xSA: eXtreme Software Analytics

http://www.continuinged.ku.edu/programs/ase/tutorials.php
https://sites.google.com/site/xsoftanalytics/
Thank you!

Q&A

Text Analytics for Software Engineering Bibliography
https://sites.google.com/site/text4se/

Mining Software Engineering Data Bibliography
https://sites.google.com/site/asergrp/dmse